Received 4 August 2020; | Revised 30 December 2021; | Accepted 14 May 2021

DOI: 10.1002/oca.2750

RESEARCH ARTICLE

Inverse dynamics-based formulation of finite horizon optimal control problems for rigid-body systems

Sotaro Katayama | Toshiyuki Ohtsuka*

[1]Department of System Science, Graduate School of Informatics, Kyoto University, Kyoto, Japan

Correspondence
Toshiyuki Ohtsuka, Graduate School of Informatics, Kyoto University, Kyoto, Japan.
Email: ohtsuka@i.kyoto-u.ac.jp

Summary

We propose a formulation of the finite horizon optimal control problem (FHOCP) based on inverse dynamics for general open-chain rigid-body systems, which reduces the computational cost from the conventional formulation based on forward dynamics. We regard the generalized acceleration as a decision variable and inverse dynamics as an equality constraint. To treat under-actuated systems with inverse dynamics that are well-defined only to fully actuated systems, i.e., to consider passive joints in this FHOCP, we add an equality constraint to zero the corresponding generalized torques. We include the contact forces in the decision variables of this FHOCP and treat the contact constraints using Baumgarte's stabilization method for numerical stability. We derive the optimality conditions and formulate the two-point boundary-value problem that can be efficiently solved using the recursive Newton-Euler algorithm (RNEA) and the partial derivatives of RNEA. We conducted three numerical experiments on model predictive control based on the proposed formulation to demonstrate its effectiveness. The first experiment involved simulating a swing-up control of a four-link arm with a passive joint and showed that the proposed formulation is effective for under-actuated systems. The second one involved comparing the proposed formulation with the conventional forward-dynamics-based formulation with various numbers of joints and showed that the proposed formulation reduces computational cost regardless of the number of joints. The third experiment involved simulating a whole-body control of a quadruped robot, a floating-base system having four contacts with the ground, and showed that the proposed formulation is applicable even for floating-base systems with contacts.

KEYWORDS:
optimal control, model predictive control, robotics, rigid-body systems

1 | INTRODUCTION

Nonlinear model predictive control (NMPC)[1] is a promising method for online motion planning and automatic control of nonlinear systems with constraints. It generates complex and efficient motion robustly against disturbances taking constraints into account by solving a finite horizon optimal control problem (FHOCP) based on the current system's state at each sampling time. However, we cannot achieve NMPC unless we can solve an FHOCP within a given sampling period. From this viewpoint, we

still need to reduce the computational cost of NMPC when we apply it to complicated rigid-body systems such as legged robots that have a huge dimensional state and complicated dynamics.

Most efficient algorithms solving an FHOCP for NMPC are gradient-descent-type methods[2] or Newton-type methods[3,4,5,6,7]. Both types of methods compute a function representing the system's dynamics and its partial derivatives to simulate the system's behavior and evaluate its sensitivities over the horizon. In an FHOCP for a rigid-body system, the state is composed of the configuration and generalized velocity, and the behavior of the system's dynamics for given decision variables is simulated over the horizon by integrating the generalized acceleration. Previous studies applying NMPC to such systems[8,9,10,11,12,13,14] assumed generalized torques as decision variables of an FHOCP and computed the generalized acceleration from the given configuration, generalized velocity, and generalized torques, which is called forward dynamics. These studies also computed the partial derivatives of the function of forward dynamics with respect to the configuration, generalized velocity, and generalized torques to apply the aforementioned algorithms of NMPC. The function of forward dynamics and its partial derivatives are so complicated that its computations occupy most of the total computational cost of NMPC, requiring efficient numerical methods to compute them. The articulated body algorithm (ABA)[15,16] is a highly efficient recursive algorithm that computes forward dynamics of open-chain rigid-body systems with degrees of freedom as large as real robots such as manipulators and humanoid robots. When it comes to the partial derivatives of the function of forward dynamics, a recursive algorithm that computes analytical derivatives of the function of forward dynamics[17] is faster than other methods such as the finite-difference approximation or automatic differentiation[18]. However, these algorithms still take much computational time, which makes it difficult to achieve NMPC.

An alternative representation of the dynamics of rigid-body systems is inverse dynamics, which is a calculation of the generalized torques for the given configuration, generalized velocity, and generalized acceleration. Like forward dynamics, inverse dynamics and the partial derivatives of the function of inverse dynamics are complicated and require efficient numerical methods. The recursive Newton Euler algorithm (RNEA)[16] is the most efficient algorithm to compute the inverse dynamics of open-chain and fully actuated rigid-body systems. To calculate the partial derivatives of the function of inverse dynamics, a recursive algorithm that computes its analytical derivatives[17], referred to as the partial derivatives of RNEA in the paper, uses sparsity and is more efficient than other methods such as finite-difference and automatic differentiation[18]. Featherstone[16] demonstrated through analysis of arithmetic operations and numerical experiments that the computational cost of RNEA is less than that of ABA. The partial derivatives of RNEA are also faster than those of the recursive algorithm for the analytical derivatives of the function of forward dynamics[17]. Therefore, we expect to reduce the computational cost by replacing forward dynamics in an FHOCP with inverse dynamics.

Previous studies on FHOCPs based on inverse dynamics were conducted in the context of direct trajectory optimization with contacts[19,20]. However, these studies focused on the stable solution method for the complementarity problem arising from contacts with the environment rather than computational efficiency. As a result, they used direct multiple shooting, in which all variables are treated as the decision variables of the optimization problem. Furthermore, they did not use efficient algorithms for rigid-body dynamics such as RNEA. In contrast, our previous study[21] proposed a reformulation of the FHOCP for fully actuated rigid-body systems by using inverse dynamics. In that study, we formulated a two-point boundary-value problem (TPBVP) that can be solved efficiently using RNEA and the partial derivatives of RNEA. We showed that the computational cost of the proposed formulation of this FHOCP based on inverse dynamics is faster than the conventional formulation based on forward dynamics through numerical experiments. In formulating this FHOCP, we argued that forward dynamics and inverse dynamics are equivalent constraints derived from the same equation of the motion if the system is fully actuated. However, we did not consider under-actuated systems and external forces, which are necessary for practical robotic applications such as systems having a floating base and contacts with the environment.

In this study, we formulated the FHOCP based on inverse dynamics for general open-chain rigid-body systems including under-actuated systems and systems having contacts with the environment. We regard the generalized acceleration as decision variables and inverse dynamics as an equality constraint in formulating this FHOCP. To treat under-actuated systems with RNEA that is originally designed to be applied only to fully actuated systems, that is, to consider the passive joints including a virtual joint between a floating base and the world frame in an FHOCP with RNEA, we add an equality constraint to zero the corresponding generalized torques. We include the contact forces in decision variables of this FHOCP and treat the contact constraints using Baumgarte's stabilization method[22]. We derive the necessary conditions of the optimal control, namely, optimality conditions, and formulate a TPBVP that can be efficiently solved using RNEA and its partial derivatives.

Note that another approach to treat contact forces is to model them as spring-damper systems[12,19]. However, this approach requires troublesome tunings of parameters that are not based on physical characteristics. Its stiffness also makes numerical

computation unstable and requires small discretization steps on the horizon, which increases the computational cost. In contrast, we incorporate the contact forces and contact constraints explicitly in an FHOCP and avoid the above drawbacks.

We conducted three numerical experiments on NMPC to demonstrate the effectiveness of the proposed formulation. The first experiment involved simulating the swing-up control of a four-link arm with a passive joint. It showed that the proposed formulation can control under-actuated systems with high nonlinearity. The second experiment involved comparing the proposed formulation based on inverse dynamics and the conventional formulation based on forward dynamics in terms of computational time. It verified that the proposed formulation reduces computational cost regardless of the numbers of actuated and passive joints. The third experiment involved simulating a whole-body control of a quadruped robot, a floating-base system with four contacts with the ground, showing that the proposed formulation is applicable to floating-base systems and systems with contacts.

This paper is composed as follows. Section 2 introduces the kinematics and dynamics of rigid-body systems and algorithms to compute rigid-body dynamics and the partial derivatives of the functions of rigid-body dynamics. Section 3 introduces the proposed formulation of an FHOCP based on inverse dynamics, derives the optimality conditions, and formulates a TPBVP. Section 4 explains the numerical experiments of NMPC we conducted and the effectiveness of the proposed formulation. Section 5 concludes our paper.

2 | RIGID-BODY SYSTEMS

2.1 | Configuration of rigid-body systems

The configuration of a rigid-body system lies in a differentiable manifold (Lie group), and the generalized velocity lies in its tangent space (Lie algebra)[23]. The representations of the configuration and velocity are classified based on the types of joints. For example, the configuration of a revolute joint lies in S^1 and its tangent space is $\mathbb{R}$. Therefore, a rigid-body system with n_J revolute joints and a fixed base, i.e., the base frame is fixed to the world frame, the configuration lies in S^{n_J} and the generalized velocity is parameterized by $\mathbb{R}^{n_J}$. In most robotic applications, however, there are limitations in the joint angles, and the configuration of the rigid-body system with a fixed base lies in Euclidean space $\mathbb{R}^{n_J}$. On the other hand, the configuration of a floating base joint, a virtual joint between the floating base and world frame, lies in SE(3) and its tangent space is se(3), which is parametrized by $\mathbb{R}^6$. Therefore, if a system has n_J revolute joints and a floating base, its configuration lies in $\mathrm{SE}(3)\times\mathbb{R}^{n_J}$, and the generalized velocity is parameterized by $\mathbb{R}^{n_J+6}$[24]. Alternatively, we can model the position of the floating base by $\mathbb{R}^3$ and its rotation by Euler angles $\mathbb{R}^3$. In this case, the configuration of the system with n_J revolute joints and a floating base then lies in $\mathbb{R}^{n_J+6}$, and the generalized velocity is parameterized by $\mathbb{R}^{n_J+6}$. However, we may suffer from singularities in the base frame rotation represented by Euler angles. Therefore, for such systems, modeling based on the differentiable manifold, i.e., Lie group, is preferable to Euler angles. In the following, we formulate an FHOCP with the configuration taking an arbitrary form, e.g., SE(3) and Euler angles.

To describe the evolution of the configuration on a differentiable manifold Q uniformly, we introduce an addition operator $\oplus$ and subtraction operator $\ominus$ on Q. Suppose that the generalized velocity lies in $\mathbb{R}^n$. The evolution of the configuration q with its tangential generalized velocity $v \in \mathbb{R}^n$ in time δt is then described as $q \oplus v\delta t \in Q$. The difference of the two configurations $q_1, q_2 \in Q$ is described as $q_1 \ominus q_2 \in \mathbb{R}^n$. For example, if the configuration lies in the Euclidean space $\mathbb{R}^n$, $\oplus$ and $\ominus$ are just given by the addition and subtraction on the Euclidean space. If the configuration lies in SE(3), then $\oplus$ and $\ominus$ apply the exponential map and logarithmic map[25], respectively. For notational convenience, we further introduce a function denoting the subtraction $\delta(\cdot,\cdot) : Q \times Q \to \mathbb{R}^n$ such that

$$\delta(q^+, q^-) := q^+ \ominus q^-. \tag{1}$$

We describe the partial derivative of (1) with respect to the first argument as $\frac{\partial \delta}{\partial q^+}(\cdot,\cdot)$ and that with respect to the second argument as $\frac{\partial \delta}{\partial q^-}(\cdot,\cdot)$.

2.2 | Dynamics of rigid-body systems

Suppose that Q is a differentiable manifold in which the configuration of the rigid-body system lies, its tangent space is parameterized by $\mathbb{R}^{n_v}$ ($n_v > 0$), and the dimension of the generalized torques of the system is given by n_A ($n_v \geq n_A > 0$). We also assume that the system has $N_c \geq 0$ contact points with the environment and each contact imposes a $n_{c,i}$ ($6 \geq n_{c,i} > 0$) dimensional constraint on $q \in Q$, i.e., $\psi_i(q) = 0$, where $\psi_i(\cdot) : Q \to \mathbb{R}^{n_{c,i}}$. For example, a point contact, which imposes a constraint on the position of the end-effector, has $n_{c,i} = 3$, and the surface contact, which constraints on the position and rotation of the end-effector, has $n_{c,i} = 6$. We define the total dimension of the contact constraints as $n_c := \sum_{i=1}^{N_c} n_{c,i}$. For each contact, contact

force $f_i^{\text{ext}} \in \mathbb{R}^{n_{c,i}}$ works in the system through the i-th contact point. The equation of the motion for the configuration $q \in Q$, parameterization of the generalized velocity $v \in \mathbb{R}^{n_v}$, and generalized torques $u \in \mathbb{R}^{n_A}$ is written as

$$M(q)\dot{v} + h(q, v) = \sum_{i=1}^{N_c} J_i^{\mathrm{T}}(q) f_i^{\text{ext}} + S^{\mathrm{T}} u, \tag{2}$$

where $M(q) \in \mathbb{R}^{n_v \times n_v}$ denotes the joint-space inertia matrix, $h(q, h) \in \mathbb{R}^{n_v}$ denotes Coriolis, centrifugal, and gravitational terms, $S \in \mathbb{R}^{n_A \times n_v}$ denotes the selection matrix that extracts actuated configuration, and $J_i(q) \in \mathbb{R}^{n_{c,i} \times n_v}$ is the Jacobian matrix of the i-th contact constraint[26]. The function of forward dynamics, which we describe as $\mathrm{FD}(\cdot, \cdot, \cdot, \cdot) : Q \times \mathbb{R}^{n_v} \times \mathbb{R}^{n_A} \times \mathbb{R}^{n_c} \to \mathbb{R}^{n_v}$, is defined as

$$\dot{v} = \mathrm{FD}(q, v, u, f) = M(q)^{-1} \left\{ S^{\mathrm{T}} u + \sum_{i=1}^{N_c} J_i^{\mathrm{T}}(q) f_i^{\text{ext}} - h(q, v) \right\}, \tag{3}$$

where $f \in \mathbb{R}^{n_c}$ denotes the stack of the external forces f_i^{ext}. Inverse dynamics is well-defined only for fully actuated systems, i.e., for systems with $n_A = n_v$. The function of inverse dynamics of a fully actuated system, which we describe as $\mathrm{ID}(\cdot, \cdot, \cdot, \cdot) : Q \times \mathbb{R}^{n_v} \times \mathbb{R}^{n_v} \times \mathbb{R}^{n_c} \to \mathbb{R}^{n_v}$, is explicitly defined as

$$u = \mathrm{ID}(q, v, \dot{v}, f) = M(q)\dot{v} + h(q, v) - \sum_{i=1}^{N_c} J_i^{\mathrm{T}}(q) f_i^{\text{ext}}, \tag{4}$$

where we assume $S = I$ without loss of generality.

2.3 | Rigid-body dynamics algorithms

Both forward dynamics $\mathrm{FD}(q, v, u, f)$ and inverse dynamics $\mathrm{ID}(q, v, \dot{v}, f)$ are complex and difficult to derive explicitly from the equations of motion (2) when the system has a large number of joints. ABA[15,16] is one of the most efficient algorithms to compute $\mathrm{FD}(q, v, u, f)$ for open-loop rigid-body systems, and RNEA[16] is the fastest algorithm to compute $\mathrm{ID}(q, v, \dot{v}, f)$ for open-loop fully actuated rigid-body systems. We hereafter use these algorithm to compute $\mathrm{FD}(q, v, u, f)$ and $\mathrm{ID}(q, v, \dot{v}, f)$. The computational complexities of ABA and RNEA are both $O(n)$. However, Featherstone showed that RNEA is faster than ABA through analysis of the arithmetic operations and numerical experiments[16].

Most efficient algorithms to solve FHOCPs for NMPC use the gradient and Hessian of the cost function and require the partial derivatives of the function of the system's dynamics. We therefore need to compute the partial derivatives of $\mathrm{FD}(q, v, u, f)$ or those of $\mathrm{ID}(q, v, \dot{v}, f)$ in solving this FHOCP. However, explicit derivations of the partial derivatives of $\mathrm{FD}(q, v, u, f)$ and $\mathrm{ID}(q, v, \dot{v}, f)$ are also difficult because of their complexities. To solve this problem, efficient recursive algorithms using the sparsity structure have been proposed[17]. The authors proposed a fast algorithm to compute the partial derivatives of $\mathrm{ID}(q, v, \dot{v}, f)$, called the partial derivatives of RNEA. They also found that the partial derivatives of $\mathrm{FD}(q, v, u, f)$ can be computed by multiplying $M^{-1}(q)$ to the partial derivatives of $\mathrm{ID}(q, v, \dot{v}, f)$. Because of the origin of these algorithms, the computational cost of the partial derivatives of $\mathrm{ID}(q, v, \dot{v}, f)$ takes less computational time than that of their algorithm for the partial derivatives of $\mathrm{FD}(q, v, u, f)$.

3 | INVERSE DYNAMICS-BASED OPTIMAL CONTROL PROBLEMS

3.1 | Formulation of the optimal control problems

In this section, we formulate the FHOCP for general open-chain rigid-body systems based on inverse dynamics. We derive the necessary conditions of the optimal control and formulate a TPBVP. As in the previous section, Q is the differentiable manifold in which the configuration of the system lies and the generalized velocity is parameterized by $\mathbb{R}^{n_v}$. The state is then composed by $q \in Q$ and $v \in \mathbb{R}^{n_v}$. Let $[t_0, t_f]$ $(t_0 < t_f)$ be the time interval in which the FHOCP is defined. To formulate the FHOCP based on inverse dynamics, we treat the generalized acceleration $\dot{v}(\tau) \in \mathbb{R}^{n_v}$ $(t_0 \leq \tau < t_f)$ and stack of contact forces $f(\tau) \in \mathbb{R}^{n_c}$ $(t_0 \leq \tau < t_f)$ as the decision variables. We assume that the system can be under-actuated. However, $\mathrm{ID}(q, v, \dot{v}, f)$ cannot be explicitly defined for under-actuated systems. Therefore, we cannot directly apply RNEA and its partial derivatives to such systems. To tackle this problem, we formulate the FHOCP with $\mathrm{ID}(q, v, \dot{v}, f)$ defined for fully actuated systems and introduce an equality constraint that zeros the generalized torques corresponding to the passive joints. For example, we assume that even a

floating-base joint, the virtual 6-degrees of freedom (DOF) joint that connects the floating base and world frame, can generate 6-dimensional torques in the FHOCP. To take into account that floating-base joints are passive, we introduce an equality constraint to zero these 6-dimensional torques in the FHOCP.

Since $\dot{v}(\tau)$ $(t_0 \leq \tau < t_f)$ is given, the state transition is simply described by

$$\begin{bmatrix} q(\tau + \delta\tau) \\ v(\tau + \delta\tau) \end{bmatrix} = \begin{bmatrix} q(\tau) \oplus v(\tau)\delta\tau \\ v(\tau) + \dot{v}(\tau)\delta\tau \end{bmatrix}, \tag{5}$$

where $\delta\tau > 0$ is the infinitesimal time increment. Equation (5) is also transformed into

$$\begin{bmatrix} \delta(q(\tau + \delta\tau), q(\tau)) \\ v(\tau + \delta\tau) - v(\tau) \end{bmatrix} - \begin{bmatrix} v(\tau)\delta\tau \\ \dot{v}(\tau)\delta\tau \end{bmatrix} = 0. \tag{6}$$

To consider the dynamics of the system (2), we introduce an equality constraint

$$\mathrm{ID}(q(\tau), v(\tau), \dot{v}(\tau), f(\tau)) - u(\tau) = 0, \tag{7}$$

where $u(\tau) \in \mathbb{R}^{n_v}$ $(t_0 \leq \tau < t_f)$ denotes the generalized torques for the fully actuated system and $\mathrm{ID}(q(\tau), v(\tau), \dot{v}(\tau), f(\tau))$ takes the form of inverse dynamics for the fully actuated system as in (4). We introduce the equality constraints to zero the generalized torques corresponding to the passive joints, i.e.,

$$u_{(i)}(\tau) = 0, \ i \in \bar{\mathscr{I}}, \tag{8}$$

where $u_{(i)}$ denotes the i-th component of $u(\tau)$ and $\bar{\mathscr{I}}$ denotes the set of indices of generalized torques corresponding to the passive joints. We introduce a passive-selection matrix $\bar{S} \in \mathbb{R}^{n_v \times n_P}$, where $n_P = n_v - n_A$, which extracts the generalized torques corresponding to passive joints. We then consider the equality constraint

$$\bar{S}^{\mathrm{T}} u(\tau) = 0 \tag{9}$$

instead of constraints (8). We also assume that the system has N_c contacts with the environment, denoted as $\psi_i(q) = 0$ for $i = 1, ..., N_c$. We then consider the constraint

$$\psi(q(\tau)) = 0, \ \ t_0 \leq \tau < t_f, \tag{10}$$

where $\psi(\cdot) : Q \to \mathbb{R}^{n_c}$ is the stack of contact constraints $\psi_i(q)$ for $i = 1, ..., N_c$ in the FHOCP. If the initial configuration $q(t)$ satisfies $\psi(q(t)) = 0$, the contact constraint on the configuration (10) can be transformed into that on the configuration and generalized velocity, i.e.,

$$\psi_v(q(\tau), v(\tau)) := \frac{d}{d\tau}\psi(q(\tau)) = \frac{\partial \psi}{\partial q}(q(\tau))v(\tau) = J_C(q(\tau))v(\tau) = 0, \ \ t_0 \leq \tau < t_f, \tag{11}$$

where $J_C(q) := \frac{\partial \psi}{\partial q}(q)$ is equivalent to the stack of the contact Jacobians $\frac{\partial \psi_i}{\partial q}(q)$ for $i = 1, ..., N_c$. If the initial configuration $q(t)$ and velocity $v(t)$ satisfy $\psi(q(t)) = 0$ and $\psi_v(q(t), v(t)) = 0$, constraint (11) can be further transformed into the constraint on the configuration, generalized velocity, and generalized acceleration, i.e.,

$$\psi_a(q(\tau), v(\tau), \dot{v}(\tau)) := \frac{d}{d\tau}\psi_v(q(\tau), v(\tau)) = \dot{J}_C(q(\tau), v(\tau))v(\tau) + J_C(q(\tau))\dot{v}(\tau) = 0, \ \ t_0 \leq \tau < t_f.$$

In this study, we used Baumgarte's stabilization method[22] for numerical stability. We consider the constraint

$$\Psi(q(\tau), v(\tau), \dot{v}(\tau)) := \psi(q(\tau)) + k_v\psi_v(q(\tau), v(\tau)) + k_q\psi_a(q(\tau), v(\tau), \dot{v}(\tau)) = 0, \ \ t_0 \leq \tau < t_f, \tag{12}$$

instead of the original constraint (10) in the FHOCP, where $k_v \geq 0$ and $k_q \geq 0$ are stabilization parameters. The FHOCP for time interval $[t_0, t_f]$ is then given as follows: find the optimal generalized acceleration $\dot{v}(\tau)$ $(t_0 \leq \tau < t_f)$ and the stack of contact forces $f(\tau)$ $(t_0 \leq \tau < t_f)$ minimizing the cost function

$$J = \varphi(t_f, q(t_f), v(t_f)) + \int_{t_0}^{t_f} L(\tau, q(\tau), v(\tau), \dot{v}(\tau), u(\tau), f(\tau))d\tau, \tag{13}$$

where $\varphi(\cdot, \cdot, \cdot) : \mathbb{R} \times Q \times \mathbb{R}^{n_v} \to \mathbb{R}$ denotes the terminal cost and $L(\cdot, \cdot, \cdot, \cdot, \cdot, \cdot) : \mathbb{R} \times Q \times \mathbb{R}^{n_v} \times \mathbb{R}^{n_v} \times \mathbb{R}^{n_v} \times \mathbb{R}^{n_c} \to \mathbb{R}$ denotes the stage cost, subject to (6), (7), (9), (12), and an equality constraint

$$C(\tau, q(\tau), v(\tau), \dot{q}(\tau), u(\tau), f(\tau)) = 0, \tag{14}$$

where $C(\cdot, \cdot, \cdot, \cdot, \cdot, \cdot) : \mathbb{R} \times Q \times \mathbb{R}^{n_v} \times \mathbb{R}^{n_v} \times \mathbb{R}^{n_v} \times \mathbb{R}^{n_c} \to \mathbb{R}^m$.

For numerical computation, we discretize the FHOCP. We divide the time interval $[t_0, t_f]$ into N steps by introducing $t_i := t + i\Delta\tau$ for $i = 0, ..., N$, where $\Delta\tau := (t_f - t_0)/N$. We discretize the configuration, parameterization of the generalized velocity, generalized acceleration, generalized torques, and contact forces as $q_0, ..., q_N \in Q$, $v_0, ..., v_N \in \mathbb{R}^{n_v}$, $a_0, ..., a_{N-1} \in \mathbb{R}^{n_v}$, $u_0, ..., u_{N-1} \in \mathbb{R}^{n_v}$, $f_0, ..., f_{N-1} \in \mathbb{R}^{n_c}$, respectively. The discretized FHOCP is then given as follows: find a_0, ..., a_{N-1} and f_0, ..., f_{N-1} minimizing the cost function

$$J = \varphi(t_N, q_N, v_N) + \sum_{i=0}^{N-1} L(t_i, q_i, v_i, a_i, u_i, f_i)\Delta\tau, \tag{15}$$

subject to

$$q_0 = q(t_0),\ v_0 = v(t_0), \tag{16}$$

$$v_i\Delta\tau - \delta(q_{i+1}, q_i) = 0,\ i = 0, ..., N-1, \tag{17}$$

$$a_i\Delta\tau + v_i - v_{i+1} = 0,\ i = 0, ..., N-1, \tag{18}$$

$$\mathrm{ID}(q_i, v_i, a_i, f_i) - u_i = 0,\ i = 0, ..., N-1, \tag{19}$$

$$\Psi(q_i, v_i, a_i) = 0,\ i = 0, ..., N-1, \tag{20}$$

$$\bar{S}^{\mathrm{T}} u_i = 0,\ i = 0, ..., N-1, \tag{21}$$

and

$$C(t_i, q_i, v_i, a_i, u_i, f_i) = 0,\ i = 0, ..., N-1. \tag{22}$$

The augmented cost function is then given by

$$\begin{aligned}\tilde{J} = {} & \varphi(t_N, q_N, v_N) + \sum_{i=0}^{N-1} L(t_i, q_i, v_i, a_i, u_i, f_i)\Delta\tau + \sum_{i=0}^{N-1} \lambda_{i+1}^{\mathrm{T}}(v_i\Delta\tau - \delta(q_i, q_{i+1})) + \sum_{i=0}^{N-1} \gamma_{i+1}^{\mathrm{T}}(a_i\Delta\tau + v_i - v_{i+1}) \\ & + \sum_{i=0}^{N-1} \beta_i^{\mathrm{T}}(\mathrm{ID}(q_i, v_i, a_i, f_i) - u_i)\Delta\tau + \sum_{i=0}^{N-1} \eta_i^{\mathrm{T}}\Psi(q_i, v_i, a_i)\Delta\tau + \sum_{i=0}^{N-1} \nu_i^{\mathrm{T}}\bar{S}u_i\Delta\tau + \sum_{i=0}^{N-1} \mu_i^{\mathrm{T}}C(t_i, q_i, v_i, a_i, u_i, f_i)\Delta\tau,\end{aligned} \tag{23}$$

where $\lambda_1, ..., \lambda_N \in \mathbb{R}^{n_v}$, $\gamma_1, ..., \gamma_N \in \mathbb{R}^{n_v}$, $\beta_0, ..., \beta_{N-1} \in \mathbb{R}^{n_v}$, $\eta_0, ..., \eta_{N-1} \in \mathbb{R}^{n_c}$, $\nu_0, ..., \nu_{N-1} \in \mathbb{R}^{n_P}$, and $\mu_0, ..., \mu_{N-1} \in \mathbb{R}^m$ are the Lagrange multipliers with respect to (17)–(22), respectively. The optimality conditions are then derived as follows (their derivations are given in Appendix A):

$$\lambda_N = \left\{ \left(\frac{\partial \delta}{\partial q^+} \right)^{\mathrm{T}} \right\}^{-1} (q_{N-1}, q_N) \left(\frac{\partial \varphi}{\partial q} \right)^{\mathrm{T}} (t_N, q_N, v_N), \tag{24}$$

$$\begin{aligned}\lambda_i = {} & \left\{ \left(\frac{\partial \delta}{\partial q^+} \right)^{\mathrm{T}} \right\}^{-1} (q_{i-1}, q_i) \left\{ -\left(\frac{\partial \delta}{\partial q^-} \right)^{\mathrm{T}} (q_i, q_{i+1})\lambda_{i+1} + \left(\frac{\partial L}{\partial q} \right)^{\mathrm{T}} (t_i, q_i, v_i, a_i, u_i, f_i)\Delta\tau + \left(\frac{\partial \mathrm{ID}}{\partial q} \right)^{\mathrm{T}} (q_i, v_i, a_i, f_i)\beta_i\Delta\tau \right. \\ & \left. + \left(\frac{\partial \Psi}{\partial q} \right)^{\mathrm{T}} (q_i, v_i, a_i)\eta_i\Delta\tau + \left(\frac{\partial C}{\partial q} \right)^{\mathrm{T}} (t_i, q_i, v_i, a_i, u_i, f_i)\mu_i\Delta\tau \right\} = 0,\quad i = 1, ..., N-1,\end{aligned} \tag{25}$$

$$\gamma_N = \left(\frac{\partial \varphi}{\partial v} \right)^{\mathrm{T}} (t_N, q_N, v_N), \tag{26}$$

$$\begin{aligned}\gamma_i = {} & \gamma_{i+1} + \lambda_{i+1}\Delta\tau + \left(\frac{\partial L}{\partial v} \right)^{\mathrm{T}} (t_i, q_i, v_i, a_i, u_i, f_i)\Delta\tau + \left(\frac{\partial \mathrm{ID}}{\partial v} \right)^{\mathrm{T}} (q_i, v_i, a_i, f_i)\beta_i\Delta\tau + \left(\frac{\partial \Psi}{\partial v} \right)^{\mathrm{T}} (q_i, v_i, a_i)\eta_i\Delta\tau \\ & + \left(\frac{\partial C}{\partial v} \right)^{\mathrm{T}} (t_i, q_i, v_i, a_i, u_i, f_i)\mu_i\Delta\tau = 0,\quad i = 1, ..., N-1,\end{aligned} \tag{27}$$

$$\left(\frac{\partial L}{\partial \dot{v}}\right)^{\mathrm{T}}(t_i, q_i, v_i, a_i, u_i, f_i) + \gamma_{i+1} + \left(\frac{\partial \mathrm{ID}}{\partial \dot{v}}\right)^{\mathrm{T}}(q_i, v_i, a_i, f_i)\beta_i + \left(\frac{\partial \Psi}{\partial \dot{v}}\right)^{\mathrm{T}}(q_i, v_i, a_i)\eta_i + \left(\frac{\partial C}{\partial \dot{v}}\right)^{\mathrm{T}}(t_i, q_i, v_i, a_i, u_i, f_i)\mu_i = 0, \quad i = 0, ..., N-1, \tag{28}$$

$$\beta_i = \left(\frac{\partial L}{\partial u}\right)^{\mathrm{T}}(t_i, q_i, v_i, a_i, u_i, f_i) + \bar{S}^{\mathrm{T}}\nu_i + \left(\frac{\partial C}{\partial u}\right)^{\mathrm{T}}(t_i, q_i, v_i, a_i, u_i, f_i)\mu_i, \quad i = 0, ..., N-1, \tag{29}$$

and

$$\left(\frac{\partial L}{\partial f}\right)^{\mathrm{T}}(t_i, q_i, v_i, a_i, u_i, f_i) + \left(\frac{\partial \mathrm{ID}}{\partial f}\right)^{\mathrm{T}}(q_i, v_i, a_i, f_i)\beta_i + \left(\frac{\partial C}{\partial f}\right)^{\mathrm{T}}(t_i, q_i, v_i, a_i, u_i, f_i)\mu_i = 0, \ i = 0, ..., N-1. \tag{30}$$

The FHOCP is then reduced to the following nonlinear equations: find the sequence of the optimal generalized acceleration a_0, ..., a_{N-1}, contact forces f_0, ..., f_{N-1}, and Lagrange multipliers $\eta_0, ..., \eta_{N-1}$, $\nu_0, ..., \nu_{N-1}$, and $\mu_0, ..., \mu_{N-1}$, which are the decision variables of this FHOCP, satisfying (16)–(22) and (24)–(30). That is, under given $a_0, ..., a_{N-1}$, $f_0, ..., f_{N-1}$, $\eta_0, ..., \eta_{N-1}$, $\nu_0, ..., \nu_{N-1}$, and $\mu_0, ..., \mu_{N-1}$, we can first determine v_0 and q_0 from (16) and determine v_1, ..., v_N and q_1, ..., q_N from

$$v_{i+1} = v_i + a_i \Delta\tau, \quad i = 0, ..., N-1, \tag{31}$$

and

$$q_{i+1} = q_i \oplus v_i \Delta\tau, \quad i = 0, ..., N-1. \tag{32}$$

We can then determine u_0, ..., u_{N-1} from $u_i = \mathrm{ID}(q_i, v_i, a_i, f_i)$ and β_0, ..., β_{N-1} from (29). Finally, we can determine λ_N, ..., λ_1 from (24) and (25) and γ_N, ..., γ_1 from (26) and (27), which formulates the TPBVP. The errors from the optimal control of given $a_0, ..., a_{N-1}$, $f_0, ..., f_{N-1}$, $\eta_0, ..., \eta_{N-1}$, $\nu_0, ..., \nu_{N-1}$, and $\mu_0, ..., \mu_{N-1}$ are then be given by (28), (30), and (20)–(22). We can then obtain the optimal values of $a_0, ..., a_{N-1}$, $f_0, ..., f_{N-1}$, $\eta_0, ..., \eta_{N-1}$, $\nu_0, ..., \nu_{N-1}$, and $\mu_0, ..., \mu_{N-1}$ by solving the nonlinear equations, e.g., by using the gradient descent or Newton's method.

3.2 | Application to NMPC

Next, we apply the formulated TPBVP to NMPC for rigid-body systems. In NMPC, an FHOCP from the current time t to the finite future $t + T$ ($T > 0$), i.e., an FHOCP for the time interval $[t, t + T]$, is solved based on the measured or estimated state at t. We consider gradient-type methods[2,27] or the Hessian-free Newton-type methods[3,4,5,6,7], that is, we do not need to compute further derivatives of the optimality conditions (17)–(22) and (24)–(30). It is worth noting that typical efficient Newton-type methods of NMPC use Hessian approximation such as finite-difference[3] or the Gauss-Newton method[4,5,28,6]. This is because further derivatives of the optimality conditions take huge computational time and are impractical. These methods originally directly compute the optimal control input on the finite horizon $[t, t + T]$ by solving an FHOCP. The initial value of the optimal control input is then applied to the actual system. On the other hand, the solution of a FHOCP with the proposed formulation based on inverse dynamics does not explicitly contain the optimal control input $u_0, ..., u_{N-1}$. Instead, it contains the generalized acceleration $a_0, ..., a_{N-1}$ and the stack of the external force $f_0, ..., f_{N-1}$. Therefore, after computing the optimal solution, we additionally have to compute the actual control input to the system by, e.g.,

$$u(t_{\mathrm{app}}) = \mathrm{ID}(\tilde{q}(t_{\mathrm{app}}), \tilde{v}(t_{\mathrm{app}}), a_0, f_0), \tag{33}$$

where t_{app} is the instant when the optimal control input is applied to the system and $\tilde{q}(t_{\mathrm{app}})$ and $\tilde{v}(t_{\mathrm{app}})$ are the estimations of the configuration and generalized velocity at t_{app}, respectively. Note that we can compute (33) with a sufficiently short computational time compared with the entire computational burden of the FHOCP thanks to RNEA. If a time lag between the state measurement and application of the control input is sufficiently short, i.e., $t \simeq t_{\mathrm{app}}$, we can estimate $\tilde{q}(t_{\mathrm{app}}) \simeq q(t)$ and $\tilde{v}(t_{\mathrm{app}}) \simeq v(t)$. When we cannot disregard the lag, we estimate the configuration and generalized velocity using the optimal generalized acceleration computed at the previous sampling time $\hat{a}_0, ..., \hat{a}_{N-1}$ by, e.g.,

$$\tilde{q}(t_{\mathrm{app}}) = q(t) \oplus (t_{\mathrm{app}} - t)v(t), \ \tilde{v}(t_{\mathrm{app}}) = v(t) + (t_{\mathrm{app}} - t)\hat{a}_0. \tag{34}$$

4 | NUMERICAL EXPERIMENTS

We conducted three numerical experiments on NMPC based on the proposed formulation involving 1) a swing-up control of a four-link arm with a passive joint, 2) a comparison of the computational time between the proposed formulation based on inverse dynamics and the conventional formulation based on forward dynamics with various numbers of joints, and 3) a whole-body control of a quadruped robot.

Continuation/generalized minimal residual (C/GMRES) method

We used the C/GMRES method[3] as an efficient Hessian-free method of NMPC. The C/GMRES method achieves fast computation by tracking the solution of the FHOCP, i.e., it computes the time variation of the optimal solution instead of computing it directly by solving an FHOCP. Let $U(t)$ be the vector of decision variables of the FHOCP and $F(U(t), x(t), t) = 0$ be the equation that $U(t)$ has to satisfy. In the present setting, $U(t)$ is composed of $a_0, ..., a_{N-1}, f_0, ..., f_{N-1}, \eta_0, ..., \eta_{N-1}, \nu_0, ..., \nu_{N-1}, \mu_0, ..., \mu_{N-1}$ and $F(U(t), x(t), t)$ is composed of (28), (30), (20)–(22). The C/GMRES method does not directly solve the nonlinear equation $F(U(t), x(t), t) = 0$ but solves the following equation derived using the continuation method[29]

$$\frac{\partial F}{\partial U}\dot{U} = -\frac{\partial F}{\partial x}\dot{x} - \frac{\partial F}{\partial t} - \zeta F, \tag{35}$$

where $\zeta > 0$ is a stabilization parameter and typically set by the reciprocal of the sampling period. Note that we omit the arguments in (35). The products of the partial derivatives of F and vectors in (35) are computed by the forward-difference approximation of corresponding directional derivatives, and the partial derivatives of F are not computed explicitly. That is, we do not need to compute the second-order derivatives of the state equation, constraints, and cost functions, which means it is a Hessian-free method. The C/GMRES method computes $\dot{U}$ by solving linear problem (35) using the GMRES method[30], a fast inexact numerical solver of the linear problem, and updates the solution by

$$U(t + \Delta t) = U(t) + \dot{U}\Delta t, \tag{36}$$

where $\Delta t > 0$ is the sampling period.

We consider the following two C/GMRES methods of NMPC: the C/GMRES method for solving the TPBVP derived with the proposed formulation in Section 3 and that for solving the TPBVP derived with the conventional formulation based on forward dynamics, which is described in Appendix B. We call the former C/GMRES (ID) and the latter as C/GMRES (FD). For both methods, we set the length of the horizon as a time-dependent smooth function $T(t)$ such that $T(0) = 0$ and $T(t) \to T_f$ $(t \to \infty)$ as, e.g.,

$$T(t) = T_f(1 - e^{-\alpha t}), \tag{37}$$

for the initialization of the solution[3], where $T_f, \alpha > 0$. Throughout the following numerical experiments, we set the increment of the forward-difference approximation in (35) as $h = 1.0 \times 10^{-8}$ and the stabilization parameters ζ in (35) by the reciprocal of the sampling period. The remaining parameters of the C/GMRES method are the number of discretizations of the horizon N and that of iterations of the GMRES $k_{\max}$, which we do not fix here. Note that the computational time of the C/GMRES method is determined by N, $k_{\max}$, and the DOF of the system. As N increases, the computational time increases because the number of times to calculate RNEA, ABA, and partial derivatives of ID$(q, v, \dot{v}, f)$ and FD(q, v, τ, f) increases. Instead, the accuracy of the solution increases because the approximation of the continuous FHOCP by the discretized FHOCP becomes more accurate. As $k_{\max}$ increases, the computational cost also increases because the linear problem (35) is solved more accurately. As the DOF increases, the computational times of RNEA, ABA, and partial derivatives of ID$(q, v, \dot{v}, f)$ and FD(q, v, τ, f) increase. Note that C/GMRES (ID) includes the computation of (33) after updating the optimal solution by the C/GMRES method. Therefore, we measured the computational time of C/GMRES (ID) as the sum of calculations of the C/GMRES method and (33) in the following numerical experiments.

Software implementation

The C/GMRES method was implemented in C++ in the following simulations. To implement RNEA, ABA, the partial derivatives of RNEA, and the analytical derivatives of FD(q, v, τ, f), we used Pinocchio[31,32], an efficient C++ library for the rigid-body dynamics algorithms. We also used Pinocchio to compute the integration on a Lie group, partial derivatives of $\delta(q^+, q^-)$ in (24) and (25), kinematics of the rigid-body systems for contact constraint (20), and its partial derivatives in (25), (27), (28), and (30).

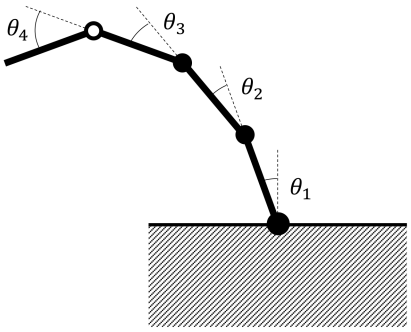

FIGURE 1 Fixed-base under-actuated four-link arm. White circle is passive joint

4.1 | Swing-up control of four-link arm with passive joint

Problem settings

We first simulated the swing-up control of the four-link arm with a passive joint which depicted in Fig. 1 to show that the proposed formulation is applicable to under-actuated systems with high nonlinearity. In this figure, black circles are the actuated joints and the white circle is the passive joint. The configuration is given by $q = [\theta_1\ \theta_2\ \theta_3\ \theta_4]^{\mathrm{T}}$, and the passive-selection matrix is given by $\bar{S} = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^{\mathrm{T}}$. We assume that each joint has no mass and no inertia and that each link has the same physical characteristics. We set the length of each link to 1m, width to 0.1m, and mass to 1kg. We also assumed that the mass is distributed uniformly in the link. We then constructed the terminal cost $\varphi(\tau, q, v)$ in (15) as

$$\varphi(\tau, q, v) = \frac{1}{2} \left(q - q_{\mathrm{ref}}\right)^{\mathrm{T}} Q_q \left(q - q_{\mathrm{ref}}\right) + \frac{1}{2} v^{\mathrm{T}} Q_v v, \tag{38}$$

where $q_{\mathrm{ref}} = [0, 0, 0, 0]^{\mathrm{T}}$, $Q_q = 1.0 \times I_4$, and $Q_v = 0.1 \times I_4$. Note that I_n denotes a $n \times n$ identity matrix. We set the stage cost $L(t, q, v, a, u)$ in (15) for C/GMRES (ID) as

$$L(t, q, v, a, u) = \frac{1}{2} \left(q - q_{\mathrm{ref}}\right)^{\mathrm{T}} Q_q \left(q - q_{\mathrm{ref}}\right) + \frac{1}{2} v^{\mathrm{T}} Q_v v + \frac{1}{2} a^{\mathrm{T}} Q_a a + \frac{1}{2} u^{\mathrm{T}} R u, \tag{39}$$

where $Q_a = 0.005 \times I_4$ and $R = 10^{-4} \times I_4$. Note that in C/GMRES (FD), we consider the penalty on the generalized acceleration as

$$\frac{1}{2\Delta\tau^2} \left(v_{i+1} - v_i\right)^{\mathrm{T}} Q_a \left(v_{i+1} - v_i\right), \tag{40}$$

which corresponds to the original penalty $\frac{1}{2} {a_i}^{\mathrm{T}} Q_a a_i$ in (39). The resultant FHOCP is described in Appendix B.

We set the parameters of both C/GMRES (FD) and C/GMRES (ID) as $N = 30$, $k_{\max} = 5$, $T_f = 1.5$, and $\alpha = 1.0$ and set the sampling period by 1 ms. Note that there are differences between the two solution methods in terms of the dimensions of the decision variables. C/GMRES (ID) computes the four-dimensional control input and one-dimensional constraint that forces the torque on the passive joint to zero in the FHOCP. The dimension of the decision variables of C/GMRES (ID) is then given by $5N$. In contrast, C/GMRES (FD) computes the three-dimensional control input in the FHOCP, and the dimension of the decision variables is given by $3N$. We also implemented NMPC for this problem with Ipopt[33], an off-the-shelf nonlinear optimization solver, to measure the baseline of the computational time. In Ipopt, we used Pinocchio to solve the FHOCP of rigid-body systems based on forward dynamics, the Broyden–Fletcher–Goldfarb–Shanno algorithm for Hessian approximation, and Harwell Subroutine Library MA57 to solve the linear subproblems of the quasi-Newton method. We set the tolerance of the errors in the optimality at each NMPC iteration to 0.005 for Ipopt.

Simulation results

Figure 2 shows the simulation results of the swing-up control of the four-link under-actuated fixed-base arm using C/GMRES (ID). The under-actuated arm successfully swung up when using C/GMRES (ID), which shows that the proposed formulation can control under-actuated systems with high nonlinearity. On the other hand, C/GMRES (FD) failed in computing even though we tested various weight parameters. Note that the computation with C/GMRES (ID) also failed without penalty on generalized acceleration $\frac{1}{2} a^{\mathrm{T}} Q_a a$. The average computational time per update of C/GMRES (ID) was 0.89 ms and that of C/GMRES (FD) was 1.37 ms on Ubuntu 18.04 LTS with the CPU Intel Core i5 2.00 GHz. C/GMRES (ID) computed the optimal solution in

real-time and was faster than C/GMRES (FD) although C/GMRES (ID) had more decision variables than C/GMRES (FD). The average computational time of Ipopt based on forward dynamics was 5.05 ms and also larger than C/GMRES (ID).

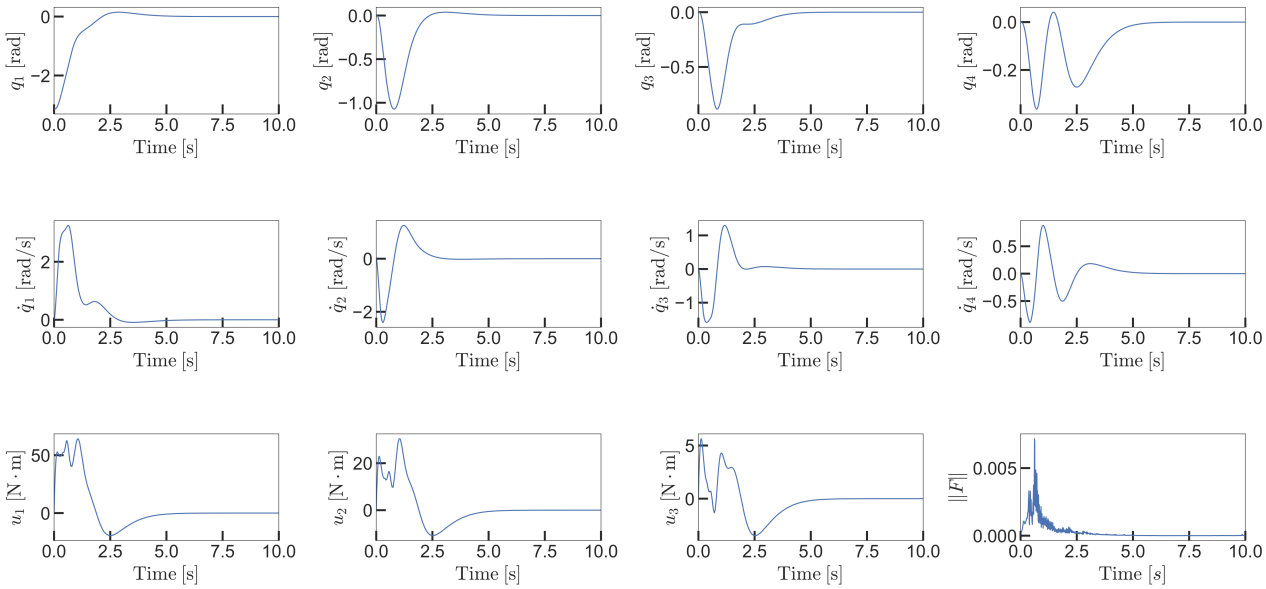

FIGURE 2 Time histories of swing-up control of four-link fixed-base arm with passive joint when $N = 30$ and $k_{\max} = 5$

4.2 | Comparison of the computational time with various numbers of joints

Problem settings

Next, we compared the computational times of C/GMRES (ID) and C/GMRES (FD) for various numbers of joints. We examined the computational time for fixed-base systems having $n_J \in \{4, 8, 16, 24, 32\}$ joints including $n_P \in \{0, 2, 4, 6, 8\}$ passive joints from the tip side. We assume that the systems do not have contacts with the environment. We set the initial configuration by $q(t) = \left[-\pi \ 0 \ \ldots \ 0\right]^{\mathrm{T}} \in \mathbb{R}^{n_J}$ and the initial velocity by $v(t) = \left[0 \ \ldots \ 0\right]^{\mathrm{T}} \in \mathbb{R}^{n_J}$. We set the cost function as the same form of (38) and (39), reference configuration as $q_{\mathrm{ref}} = \left[-\pi \ 0 \ \ldots \ 0\right]^{\mathrm{T}} \in \mathbb{R}^{n_J}$, and weight parameters in (38) and (39) as $Q_q = 1.0 \times I_n$, $Q_v = 0.1 \times I_n$, $Q_a = 0.1 \times I_n$, and $R = 0.01 \times I_n$. We set the parameters of both C/GMRES (FD) and C/GMRES (ID) as $N = 50$, $k_{\max} = 10$, $T_f = 1.0$, and $\alpha = 1.0$ and set the sampling period by 1 ms.

Experimental results

Table 1 shows the computational times per update of C/GMRES (ID) and Table 2 shows those of C/GMRES (FD). Note that "-" in Table 1 denotes that we did not consider under that condition because the number of passive joints was larger than or equal to the total number of joints, or the computations diverged as soon as the simulations started due to too many passive joints. We see that the computational time of C/GMRES (ID) decreased as much as 50% that of C/GMRES (FD) for the same n_J, n_P, N, and $k_{\max}$. From the two tables, we can see that the number of passive joints n_P had little effect on the computational times of both C/GMRES (ID) and C/GMRES (FD).

4.3 | Whole-body twisting-motion control of quadruped with contacts

Problem settings

Finally, we simulated the whole-body control of the quadruped robot ANYmal[34] using C/GMRES (ID) and verified that the proposed formulation can be used for floating-base systems with contacts. ANYmal has 12 revolute joints, a floating base, and four contacts between the tip of each of the four legs and the ground when it stands static. We model the position and

TABLE 1 Computational time [ms] per update of C/GMRES (ID) with $N = 50$ and $k_{\max} = 10$

n_J \ n_P	0	2	4	6	8
4	2.5	2.5	-	-	-
8	5.3	5.2	5.2	-	-
16	12.3	12.5	12.0	12.1	-
24	20.8	20.7	20.8	20.8	21.00
32	31.8	32.0	32.1	32.0	31.8

TABLE 2 Computational time [ms] per update of C/GMRES (FD) with $N = 50$ and $k_{\max} = 10$

n_J \ n_P	0	2	4	6	8
4	3.8	3.8	-	-	-
8	9.1	9.6	9.4	-	-
16	22.6	22.1	22.0	22.2	-
24	41.2	40.3	40.4	40.1	40.4
32	64.8	64.4	65.0	65.3	64.8

rotation of the floating base of ANYmal by SE(3). However, in designing the cost function, we model the rotation of the base frame by quaternion and represent the configuration q by a 19-dimensional tuple. For clarity, we divide q into the base frame $q_{\text{base}} := \begin{bmatrix} q_1 & q_2 & q_3 & q_4 & q_5 & q_6 & q_7 \end{bmatrix}^{\mathrm{T}}$, left-front leg $q_{\mathrm{J_{LF}}} := \begin{bmatrix} q_8 & q_9 & q_{10} \end{bmatrix}^{\mathrm{T}}$, left-hip leg $q_{\mathrm{J_{LH}}} := \begin{bmatrix} q_{11} & q_{12} & q_{13} \end{bmatrix}^{\mathrm{T}}$, right-front leg $q_{\mathrm{J_{RF}}} := \begin{bmatrix} q_{14} & q_{15} & q_{16} \end{bmatrix}^{\mathrm{T}}$, and right-hip leg $q_{\mathrm{J_{RH}}} := \begin{bmatrix} q_{17} & q_{18} & q_{19} \end{bmatrix}^{\mathrm{T}}$. Note that q_1, q_2, q_3 denotes the position of the base frame and q_4, q_4, q_6, q_7 denotes the quaternion of the rotation of the base frame. We also divide the generalized velocity $v \in \mathbb{R}^{18}$ into $v_{\text{base}} := \begin{bmatrix} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{bmatrix}^{\mathrm{T}}$, left-front leg $v_{\mathrm{J_{LF}}} := \begin{bmatrix} v_7 & v_8 & v_9 \end{bmatrix}^{\mathrm{T}}$, left-hip leg $v_{\mathrm{J_{LH}}} := \begin{bmatrix} v_{10} & v_{11} & v_{12} \end{bmatrix}^{\mathrm{T}}$, right-front leg $v_{\mathrm{J_{RF}}} := \begin{bmatrix} v_{13} & v_{14} & v_{15} \end{bmatrix}^{\mathrm{T}}$, and right-hip leg $v_{\mathrm{J_{RH}}} := \begin{bmatrix} v_{16} & v_{17} & v_{18} \end{bmatrix}^{\mathrm{T}}$. The passive-selection matrix $\bar{S}$ is given as

$$\bar{S} = \begin{bmatrix} I_6 \\ 0_{12\times 6} \end{bmatrix}, \tag{41}$$

where $0_{12\times 6} \in \mathbb{R}^{12\times 6}$ is a matrix, all elements of which are 0, and $I_6 \in \mathbb{R}^{6\times 6}$ is an identity matrix.

We control ANYmal's yaw angle to track sinusoidal reference values given by

$$\theta_{\text{ref}}(t) = \begin{cases} y_{\text{ref}} \sin \frac{2(t-t_1)\pi}{T} & (t_1 \le t \le t_1 + t_2) \\ 0 & (t < t_1, t_1 + t_2 < t) \end{cases}. \tag{42}$$

where we set $t_1 = 2$, $t_2 = 10$ and $y_{\text{ref}} = 0.12$. We construct the reference base configuration by transforming the trajectory of the Euler angle (42) into that of the quaternion. The reference trajectory is then given by $q_{\text{base,ref}}(t) := \begin{bmatrix} 0 & 0 & 0.48 & 0 & 0 & q_{6,\,\text{ref}}(t) & \sqrt{1 - q_{6,\,\text{ref}}^2(t)} \end{bmatrix}^{\mathrm{T}}$. We also construct the reference translational and angular velocity by differentiating (42) with respect to time. The reference velocity is then given by $v_{\text{base,ref}}(t) := \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \theta_{v,\,\text{ref}}(t) \end{bmatrix}^{\mathrm{T}}$, where

$$\theta_{v,\text{ref}}(t) = \begin{cases} y_{v,\text{ref}} \cos \frac{2(t-t_1)\pi}{T} & (t_1 \le t \le t_1 + t_2) \\ 0 & (t < t_1, t_1 + t_2 < t) \end{cases}, \tag{43}$$

TABLE 3 References and weight parameters of the cost function of NMPC for ANYmal. diag $\{\cdot\}$ denotes a diagonal matrix

$q_{\mathrm{J_{LF}}}$	$\begin{bmatrix}0.0315 & 0.4 & -0.806\end{bmatrix}^{\mathrm{T}}$
$q_{\mathrm{J_{LH}}}$	$\begin{bmatrix}0.0315 & -0.4 & 0.806\end{bmatrix}^{\mathrm{T}}$
$q_{\mathrm{J_{RF}}}$	$\begin{bmatrix}-0.0315 & 0.4 & -0.806\end{bmatrix}^{\mathrm{T}}$
$q_{\mathrm{J_{RH}}}$	$\begin{bmatrix}-0.0315 & -0.4 & 0.806\end{bmatrix}^{\mathrm{T}}$
$Q_{q,\mathrm{base}}$	$\mathrm{diag}\{1,\ 1,\ 10,\ 100,\ 100,\ 100,\ 100\}$
$Q_{q,\mathrm{J_{LF}}}, Q_{q,\mathrm{J_{LH}}}, Q_{q,\mathrm{J_{RF}}}, Q_{q,\mathrm{J_{RH}}}$	$0.1 \times I_3$
$Q_{v,\mathrm{base}}$	$10 \times I_6$
$Q_{v,\mathrm{J_{LF}}}, Q_{v,\mathrm{J_{LH}}}, Q_{v,\mathrm{J_{RF}}}, Q_{v,\mathrm{J_{RH}}}$	$0.1 \times I_3$
$Q_{a,\mathrm{base}}$	$0.1 \times I_6$
$Q_{a,\mathrm{J_{LF}}}, Q_{a,\mathrm{J_{LH}}}, Q_{a,\mathrm{J_{RF}}}, Q_{a,\mathrm{J_{RH}}}$	$0.01 \times I_3$
Q_u	$10^{-4} \times I_{16}$
Q_f	$10^{-4} \times I_{12}$

and $y_{v,\mathrm{ref}} = 0.15$. We then design the cost function as

$$\begin{aligned} L(t,q,v,a,u,f) = {} & \frac{1}{2}(q_{\mathrm{base}} - q_{\mathrm{base,ref}}(t))^{\mathrm{T}} Q_{q,\mathrm{base}}(q_{\mathrm{base}} - q_{\mathrm{base,ref}}(t)) + \sum_{i\in\{\mathrm{J_{LF}},\mathrm{J_{RF}},\mathrm{J_{LH}},\mathrm{J_{RH}}\}} \frac{1}{2}(q_i - q_{i,\mathrm{ref}})^{\mathrm{T}} Q_{q,i}(q_i - q_{i,\mathrm{ref}}) \\ & + \frac{1}{2}(v_{\mathrm{base}} - v_{\mathrm{base,ref}}(t))^{\mathrm{T}} Q_{v,\mathrm{base}}(v_{\mathrm{base}} - v_{\mathrm{base,ref}}(t)) + \sum_{i\in\{\mathrm{J_{LF}},\mathrm{J_{RF}},\mathrm{J_{LH}},\mathrm{J_{RH}}\}} \frac{1}{2} v_i^{\mathrm{T}} Q_{v,i} v_i \\ & + \frac{1}{2} a_{\mathrm{base}}^{\mathrm{T}} Q_{a,\mathrm{base}} a_{\mathrm{base}} + \sum_{i\in\{\mathrm{J_{LF}},\mathrm{J_{RF}},\mathrm{J_{LH}},\mathrm{J_{RH}}\}} \frac{1}{2} a_i^{\mathrm{T}} Q_{a,i} a_i + \frac{1}{2} u^{\mathrm{T}} Q_u u + \frac{1}{2} f^{\mathrm{T}} Q_f f. \end{aligned} \tag{44}$$

The other references and weight parameters are listed in Table 3. We set the parameters of C/GMRES (ID) as $N = 20$, $k_{\max} = 400$, $T_f = 0.2$, and $\alpha = 0.3$, and set the sampling period as 2.5 [ms]. We also set Baumgarte's stabilization parameters in (12) as $k_v = 20$ and $k_q = 400$. The numerical simulation was conducted on RaiSim[35,36], a physics engine for rigid-body systems with contacts.

Simulation results

Figure 3 shows snapshots of ANYmal's motion provided by raisimOgre[37], and Fig. 4 shows the time histories of the position and rotation of the base frame of ANYmal and the l_2 norm of the errors in the optimality. Note that there are too many variables in ANYmal; therefore, we show the time histories of the position and rotation of the base frame. From Fig. 4, q_6 was controlled to track $q_{6,\mathrm{ref}}(t)$, which is depicted with a dotted line, and ANYmal twisted its body sinusoidally as expected. This result indicates that the proposed formulation can be applicable to floating-base systems having contacts with the environment. The simulation showed that the errors in the optimality $\|F\|$ is large. This is because we set large Baumgarte's contact parameters k_q and k_v in (12) to stabilize numerical computation. To give details of $\|F\|$, we show the gross errors in the optimality $\|F\|$, errors in the optimality without Baumgarte's constraint (20), errors in Baumgarte's constraint (20), and errors in the original contact constraint (10) in Fig. 5. We can see that most of the errors in the optimality $\|F\|$ (solid line) came from the errors in Baumgarte's constraint (20) (dashed line) because the latter take close values of the former. Without Baumgarte's constraint, the errors in the optimality take sufficiently small values (dotted line). The errors in the original contact constraint (10) also took small values (dash-dot line), which shows that the proposed formulation can accurately take into account the contact constraint. The average computational time per update was 83 ms on Ubuntu 18.04 LTS with CPU Intel Core i7 1.80 GHz.

5 | CONCLUSION

We proposed a formulation of the FHOCP based on inverse dynamics for general open-chain rigid-body systems that reduces the computational cost compared with the conventional formulation based on forward dynamics. We regard the generalized acceleration as a decision variable and inverse dynamics as an equality constraint. To treat under-actuated systems with RNEA, which was designed to be applied only to fully actuated systems, we add an equality constraint to zero the corresponding

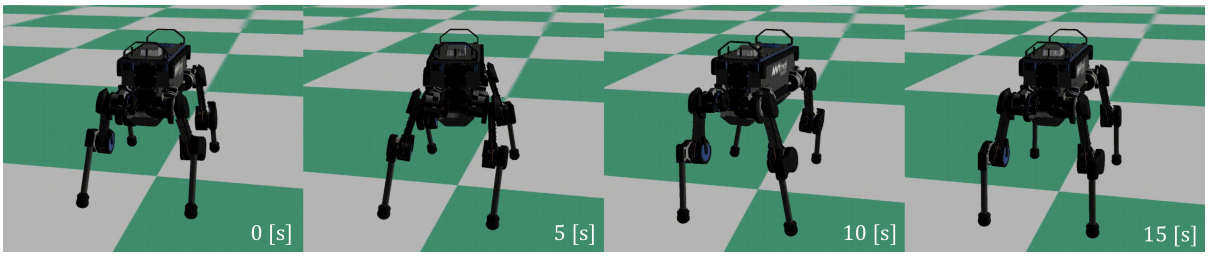

FIGURE 3 Snapshots of whole-body twisting-motion of ANYmal generated with FHOCP based on proposed formulation

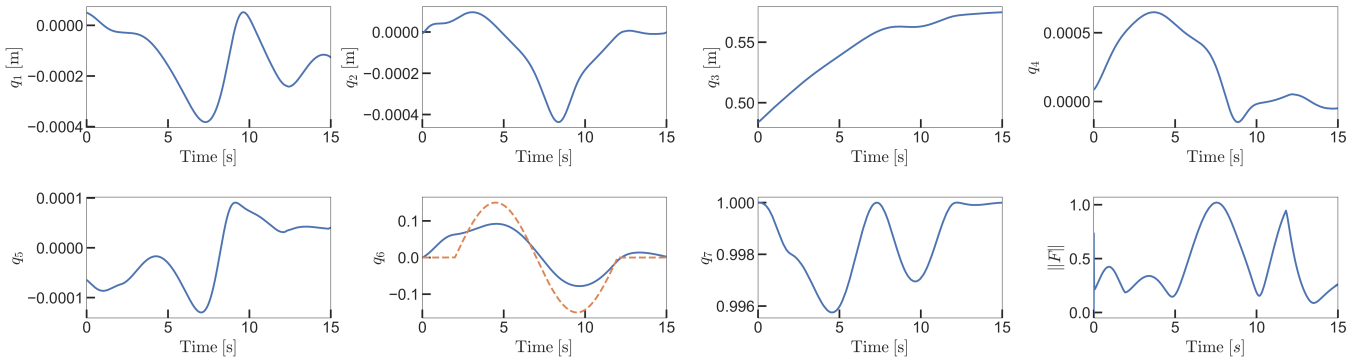

FIGURE 4 Time histories of position and rotation (quaternion) of base frame of ANYmal and l_2 norm of errors in optimality

generalized torques. We include the contact forces in the decision variables of this FHOCP and treat the contact constraints using Baumgarte's stabilization method[22]. We derive the optimality conditions and formulate a TPBVP that can be efficiently solved using RNEA and the partial derivatives of RNEA. We conducted three numerical experiments on NMPC with the C/GMRES method to show the effectiveness of the proposed formulation. In the first experiment, we simulated the swing-up control of a four-link arm with a passive joint using NMPC with the proposed formulation. Its success indicates that it can control even under-actuated systems with high nonlinearity. In the second experiment, we compared the proposed formulation based on inverse dynamics and the conventional formulation based on forward dynamics in terms of computational time for various numbers of joints. The proposed formulation reduces the computational cost by as much as 50% that of the conventional formulation regardless of the number of passive joints. In the third experiment, we simulated whole-body control of a quadruped robot, a floating-base system with four contacts with the ground. The proposed formulation is applicable to even floating-base systems with contacts.

For future work, we will consider the switches in dynamics and constraints due to contacts by combining the proposed formulation with a dedicated method[38].

How to cite this article: S. Katayama and T. Ohtsuka (2020), Inverse dynamics-based formulation of finite horizon optimal control problems for rigid-body systems.

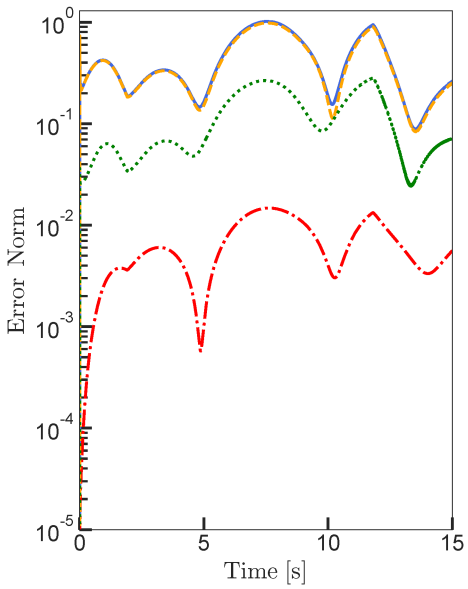

FIGURE 5 Time histories of gross errors in optimality (solid line), errors in Baumgarte's constraint (dashed line), errors in optimality without Baumgarte's constraint (dotted line), and errors in original contact constraint (dash-dot line)

APPENDIX

A DERIVATIONS OF OPTIMALITY CONDITIONS OF INVERSE-DYNAMICS-BASED FHOCP

The augmented cost function (23) is perturbed as

$$
\begin{aligned}
\delta\tilde{J} =& \frac{\partial\varphi}{\partial q}(t_N, q_N, v_N)\delta q_N + \frac{\partial\varphi}{\partial v}(t_N, q_N, v_N)\delta v_N + \sum_{i=1}^{N-1}\frac{\partial L}{\partial q}(t_i, q_i, v_i, a_i, u_i, f_i)\Delta\tau\delta q_i + \sum_{i=1}^{N-1}\frac{\partial L}{\partial v}(t_i, q_i, v_i, a_i, u_i, f_i)\Delta\tau\delta v_i \\
&+ \sum_{i=0}^{N-1}\frac{\partial L}{\partial \dot{v}}(t_i, q_i, v_i, a_i, u_i, f_i)\Delta\tau\delta a_i + \sum_{i=0}^{N-1}\frac{\partial L}{\partial u}(t_i, q_i, v_i, a_i, u_i, f_i)\Delta\tau\delta u_i + \sum_{i=0}^{N-1}\frac{\partial L}{\partial f}(t_i, q_i, v_i, a_i, u_i, f_i)\Delta\tau\delta f_i \\
&+ \sum_{i=1}^{N-1}\lambda_{i+1}^{\mathrm{T}}\delta v_i\Delta\tau - \sum_{i=1}^{N-1}\lambda_{i+1}^{\mathrm{T}}\frac{\partial\delta}{\partial q^-}(q_i, q_{i+1})\delta q_i - \sum_{i=1}^{N}\lambda_i^{\mathrm{T}}\frac{\partial\delta}{\partial q^+}(q_{i-1}, q_i)\delta q_i + \sum_{i=0}^{N-1}\gamma_{i+1}^{\mathrm{T}}\delta a_i\Delta\tau + \sum_{i=1}^{N-1}\gamma_{i+1}^{\mathrm{T}}\delta v_i \\
&- \sum_{i=1}^{N}\gamma_i^{\mathrm{T}}\delta v_i + \sum_{i=1}^{N-1}\beta_i^{\mathrm{T}}\frac{\partial \mathrm{ID}}{\partial q}(q_i, v_i, a_i, f_i)\Delta\tau\delta q_i + \sum_{i=1}^{N-1}\beta_i^{\mathrm{T}}\frac{\partial \mathrm{ID}}{\partial v}(q_i, v_i, a_i, f_i)\Delta\tau\delta v_i \\
&+ \sum_{i=0}^{N-1}\beta_i^{\mathrm{T}}\frac{\partial \mathrm{ID}}{\partial \dot{v}}(q_i, v_i, a_i, f_i)\Delta\tau\delta a_i + \sum_{i=0}^{N-1}\beta_i^{\mathrm{T}}\frac{\partial \mathrm{ID}}{\partial f}(q_i, v_i, a_i, f_i)\Delta\tau\delta f_i - \sum_{i=0}^{N-1}\beta_i^{\mathrm{T}}\delta u_i\Delta\tau \\
&+ \sum_{i=1}^{N-1}\eta_i^{\mathrm{T}}\frac{\partial\Psi}{\partial q}(q_i, v_i, a_i)\delta q_i\Delta\tau + \sum_{i=1}^{N-1}\eta_i^{\mathrm{T}}\frac{\partial\Psi}{\partial v}(q_i, v_i, a_i)\delta v_i\Delta\tau + \sum_{i=0}^{N-1}\eta_i^{\mathrm{T}}\frac{\partial\Psi}{\partial \dot{v}}(q_i, v_i, a_i)\delta a_i\Delta\tau \\
&+ \sum_{i=0}^{N-1}\nu_i^{\mathrm{T}}\bar{S}\delta u_i\Delta\tau + \sum_{i=0}^{N-1}\mu_i^{\mathrm{T}}\frac{\partial C}{\partial q}(t_i, q_i, v_i, a_i, u_i, f_i)\Delta\tau + \sum_{i=0}^{N-1}\mu_i^{\mathrm{T}}\frac{\partial C}{\partial v}(t_i, q_i, v_i, a_i, u_i, f_i)\Delta\tau + \sum_{i=0}^{N-1}\mu_i^{\mathrm{T}}\frac{\partial C}{\partial \dot{v}}(t_i, q_i, v_i, a_i, u_i, f_i)\Delta\tau \\
&+ \sum_{i=0}^{N-1}\mu_i^{\mathrm{T}}\frac{\partial C}{\partial u}(t_i, q_i, v_i, a_i, u_i, f_i)\Delta\tau + \sum_{i=0}^{N-1}\mu_i^{\mathrm{T}}\frac{\partial C}{\partial f}(t_i, q_i, v_i, a_i, u_i, f_i)\Delta\tau. \qquad \text{(A1)}
\end{aligned}
$$

The optimality conditions, the necessary conditions for $\delta\tilde{J} = 0$ under arbitrary δq_1, ..., δq_N, δv_1, ..., δv_N, δa_0, ..., δa_{N-1}, δu_0, ..., δu_{N-1}, and δf_0, ..., δf_{N-1}, are then derived by calculus of variations[39]:

$$\frac{\partial \varphi}{\partial q}(t_N, q_N, v_N) - \lambda_N^{\mathrm{T}} \frac{\partial \delta}{\partial q^+}(q_{N-1}, q_N) = 0, \tag{A2}$$

$$\frac{\partial L}{\partial q}(t_i, q_i, v_i, a_i, u_i, f_i)\Delta\tau - \lambda_{i+1}^{\mathrm{T}} \frac{\partial \delta}{\partial q^-}(q_i, q_{i+1}) - \lambda_i^{\mathrm{T}} \frac{\partial \delta}{\partial q^+}(q_{i-1}, q_i) + \beta_i^{\mathrm{T}} \frac{\partial \mathrm{ID}}{\partial q}(q_i, v_i, a_i, f_i)\Delta\tau + \eta_i^{\mathrm{T}} \frac{\partial \Psi}{\partial q}(q_i, v_i, a_i)\Delta\tau + \mu_i^{\mathrm{T}} \frac{\partial C}{\partial q}(q_i, v_i, a_i, u_i, f_i)\Delta\tau = 0, \quad i = 1, ..., N-1, \tag{A3}$$

$$\frac{\partial \varphi}{\partial v}(t_N, q_N, v_N) - \gamma_N^{\mathrm{T}} = 0, \tag{A4}$$

$$\frac{\partial L}{\partial v}(t_i, q_i, v_i, a_i, u_i, f_i)\Delta\tau + \lambda_{i+1}^{\mathrm{T}}\Delta\tau + \gamma_{i+1}^{\mathrm{T}} - \gamma_i^{\mathrm{T}} + \beta_i^{\mathrm{T}} \frac{\partial \mathrm{ID}}{\partial v}(q_i, v_i, a_i, f_i)\Delta\tau + \eta_i^{\mathrm{T}} \frac{\partial \Psi}{\partial v}(q_i, v_i, a_i)\Delta\tau + \mu_i^{\mathrm{T}} \frac{\partial C}{\partial v}(q_i, v_i, a_i, u_i, f_i)\Delta\tau = 0, \quad i = 1, ..., N-1, \tag{A5}$$

$$\frac{\partial L}{\partial \dot{v}}(t_i, q_i, v_i, a_i, u_i, f_i)\Delta\tau + \gamma_{i+1}^{\mathrm{T}}\Delta\tau + \beta_i^{\mathrm{T}} \frac{\partial \mathrm{ID}}{\partial \dot{v}}(q_i, v_i, a_i, f_i)\Delta\tau + \eta_i^{\mathrm{T}} \frac{\partial \Psi}{\partial \dot{v}}(q_i, v_i, a_i)\Delta\tau + \mu_i^{\mathrm{T}} \frac{\partial C}{\partial \dot{v}}(q_i, v_i, a_i, u_i, f_i)\Delta\tau = 0, \quad i = 0, ..., N-1, \tag{A6}$$

$$\frac{\partial L}{\partial u}(t_i, q_i, v_i, a_i, u_i, f_i)\Delta\tau - \beta_i^{\mathrm{T}}\Delta\tau + \nu_i^{\mathrm{T}}\bar{S} + \mu_i^{\mathrm{T}} \frac{\partial C}{\partial u}(q_i, v_i, a_i, u_i, f_i)\Delta\tau = 0, \ i = 0, ..., N-1, \tag{A7}$$

and

$$\frac{\partial L}{\partial f}(t_i, q_i, v_i, a_i, u_i, f_i)\Delta\tau + \beta_i^{\mathrm{T}} \frac{\partial \mathrm{ID}}{\partial f}(q_i, v_i, a_i, f_i)\Delta\tau + \mu_i^{\mathrm{T}} \frac{\partial C}{\partial f}(q_i, v_i, a_i, u_i, f_i)\Delta\tau = 0, \ i = 0, ..., N-1, \tag{A8}$$

which gives (24)–(30).

B FORWARD-DYNAMICS-BASED FHOCP

We describe the conventional formulation based on forward dynamics. We assume a system has no contact with the environment because we did not consider the contacts when comparing the proposed formulation with the conventional formulation in Section 4. Suppose that the configuration of the system lies in Q and the velocity is parameterized by $\mathbb{R}^{n_v}$. The forward dynamics is given by

$$\dot{v} = \mathrm{FD}(q, v, u) = M(q)^{-1} \left\{ S^{\mathrm{T}} u - C(q, v) \right\}, \tag{B9}$$

where $q \in Q$, $v \in \mathbb{R}^{n_v}$, and $u \in \mathbb{R}^{n_A}$. The FHOCP is then given as follows: find the optimal control input $u_0, ..., u_{N-1} \in \mathbb{R}^{n_A}$ minimizing the cost function

$$J = \varphi(t_N, q_N, v_N) + \sum_{i=0}^{N-1} L(t_i, q_i, v_i, u_i)\Delta\tau + \sum_{i=0}^{N-1} L_a(v_i, v_{i+1})\Delta\tau, \tag{B10}$$

where $\varphi(\cdot, \cdot, \cdot) : \mathbb{R} \times Q \times \mathbb{R}^{n_v} \to \mathbb{R}$ denotes the terminal cost, $L(\cdot, \cdot, \cdot, \cdot) : \mathbb{R} \times Q \times \mathbb{R}^{n_v} \times \mathbb{R}^{n_A} \to \mathbb{R}$ denotes the stage cost, and $L_a(\cdot, \cdot) : \mathbb{R}^{n_v} \times \mathbb{R}^{n_v} \to \mathbb{R}$ denotes the penalty on the acceleration, subject to

$$q_0 = q(t_0), \ v_0 = v(t_0), \tag{B11}$$

$$v_i\Delta\tau - \delta(q_{i+1}, q_i) = 0, \ i = 0, ..., N-1, \tag{B12}$$

$$\mathrm{FD}(q_i, v_i, u_i)\Delta\tau + v_i - v_{i+1} = 0, \ i = 0, ..., N-1, \tag{B13}$$

and

$$C(t_i, q_i, v_i, u_i) = 0, \ i = 0, ..., N-1, \tag{B14}$$

where $C(\cdot,\cdot,\cdot,\cdot) : \mathbb{R} \times Q \times \mathbb{R}^{n_v} \times \mathbb{R}^{n_A} \to \mathbb{m}$ denotes the equality constraint. We describe the partial derivatives of $L_a(\cdot,\cdot)$ with respect to the first argument by

$$\frac{\partial L_a}{\partial v^-}, \tag{B15}$$

and the partial derivatives of $L_a(\cdot,\cdot)$ with respect to the second argument by

$$\frac{\partial L_a}{\partial v^+}. \tag{B16}$$

The augmented cost function is given by

$$\begin{aligned}\tilde{J} = \varphi(t_N, q_N, v_N) + \sum_{i=0}^{N-1} L(t_i, q_i, v_i, u_i)\Delta\tau + \sum_{i=0}^{N-1} L_a(v_i, v_{i+1})\Delta\tau + \sum_{i=0}^{N-1} \lambda_{i+1}^{\mathrm{T}}(v_i\Delta\tau - \delta(q_i, q_{i+1})) \\ + \sum_{i=0}^{N-1} \gamma_{i+1}^{\mathrm{T}} \left\{\mathrm{FD}(q_i, v_i, u_i)\Delta\tau + v_i - v_{i+1}\right\} + \sum_{i=0}^{N-1} \mu_i^{\mathrm{T}} C(t_i, q_i, v_i, u_i)\Delta\tau,\end{aligned} \tag{B17}$$

where $\lambda_1, ..., \lambda_N \in \mathbb{R}^{n_v}$, $\gamma_1, ..., \gamma_N \in \mathbb{R}^{n_v}$, and $\mu_0, ..., \mu_{N-1} \in \mathbb{R}^m$ are the Lagrange multipliers with respect to (B12), (B13), and (B14), respectively. The augmented cost function (B17) is peturbed as

$$\begin{aligned}\delta\tilde{J} = &\frac{\partial\varphi}{\partial q}(t_N, q_N, v_N)\delta q_N + \frac{\partial\varphi}{\partial v}(t_N, q_N, v_N)\delta v_N + \sum_{i=1}^{N-1} \frac{\partial L}{\partial q}(t_i, q_i, v_i, u_i)\Delta\tau\delta q_i + \sum_{i=1}^{N-1} \frac{\partial L}{\partial v}(t_i, q_i, v_i, u_i)\Delta\tau\delta v_i \\ &+ \sum_{i=1}^{N-1} \frac{\partial L}{\partial u}(t_i, q_i, v_i, u_i)\Delta\tau\delta u_i + \sum_{i=1}^{N-1} \frac{\partial L_a}{\partial v^-}(v_i, v_{i+1})\Delta\tau\delta v_i + \sum_{i=1}^{N-1} \frac{\partial L_a}{\partial v^+}(v_i, v_{i+1})\Delta\tau\delta v_{i+1} \\ &+ \sum_{i=1}^{N-1} \lambda_{i+1}^{\mathrm{T}}\delta v_i\Delta\tau - \sum_{i=1}^{N-1} \lambda_{i+1}^{\mathrm{T}} \frac{\partial\delta}{\partial q^-}(q_i, q_{i+1})\delta q_i - \sum_{i=1}^{N} \lambda_i^{\mathrm{T}} \frac{\partial\delta}{\partial q^+}(q_{i-1}, q_i)\delta q_i \\ &+ \sum_{i=1}^{N-1} \gamma_{i+1}^{\mathrm{T}} \frac{\partial \mathrm{FD}}{\partial q}(q_i, v_i, u_i)\Delta\tau\delta q_i + \sum_{i=1}^{N-1} \gamma_{i+1}^{\mathrm{T}} \frac{\partial \mathrm{FD}}{\partial v}(q_i, v_i, u_i)\Delta\tau\delta v_i + \sum_{i=1}^{N-1} \gamma_{i+1}^{\mathrm{T}} \frac{\partial \mathrm{FD}}{\partial u}(q_i, v_i, u_i)\Delta\tau\delta u_i + \sum_{i=1}^{N-1} \gamma_{i+1}^{\mathrm{T}}\delta v_i - \sum_{i=1}^{N} \gamma_i^{\mathrm{T}}\delta v_i \\ &+ \sum_{i=0}^{N-1} \mu_i^{\mathrm{T}} \frac{\partial C}{\partial q}(t_i, q_i, v_i, u_i)\Delta\tau + \sum_{i=0}^{N-1} \mu_i^{\mathrm{T}} \frac{\partial C}{\partial v}(t_i, q_i, v_i, u_i)\Delta\tau + \sum_{i=0}^{N-1} \mu_i^{\mathrm{T}} \frac{\partial C}{\partial u}(t_i, q_i, v_i, u_i)\Delta\tau.\end{aligned} \tag{B18}$$

The optimality conditions, the necessary conditions for $\delta\tilde{J} = 0$ under arbitrary $\delta q_1, ..., \delta q_N$, $\delta v_1, ..., \delta v_N$, and $\delta u_0, ..., \delta u_{N-1}$, are then derived by calculus of variations[39] as follows:

$$\frac{\partial\varphi}{\partial q}(t_N, q_N, v_N) - \lambda_N^{\mathrm{T}} \frac{\partial\delta}{\partial q^+}(q_{N-1}, q_N) = 0, \tag{B19}$$

$$\frac{\partial L}{\partial q}(t_i, q_i, v_i, u_i)\Delta\tau - \lambda_{i+1}^{\mathrm{T}} \frac{\partial\delta}{\partial q^-}(q_i, q_{i+1}) - \lambda_i^{\mathrm{T}} \frac{\partial\delta}{\partial q^+}(q_{i-1}, q_i) + \gamma_{i+1}^{\mathrm{T}} \frac{\partial \mathrm{FD}}{\partial q}(q_i, v_i, u_i)\Delta\tau + \mu_i^{\mathrm{T}} \frac{\partial C}{\partial q}(t_i, q_i, v_i, u_i)\Delta\tau = 0, \quad i = 1, ..., N-1, \tag{B20}$$

$$\frac{\partial\varphi}{\partial v}(t_N, q_N, v_N) + \frac{\partial L_a}{\partial v^+}(v_{N-1}, v_N)\Delta\tau - \gamma_N^{\mathrm{T}} = 0, \tag{B21}$$

$$\begin{aligned}\frac{\partial L}{\partial v}(t_i, q_i, v_i, u_i)\Delta\tau + \frac{\partial L_a}{\partial v^-}(v_i, v_{i+1})\Delta\tau + \frac{\partial L_a}{\partial v^+}(v_{i-1}, v_i)\Delta\tau + \lambda_{i+1}^{\mathrm{T}}\Delta\tau + \gamma_{i+1}^{\mathrm{T}} - \gamma_i^{\mathrm{T}} + \gamma_{i+1}^{\mathrm{T}} \frac{\partial \mathrm{FD}}{\partial v}(q_i, v_i, u_i)\Delta\tau \\ + \mu_i^{\mathrm{T}} \frac{\partial C}{\partial v}(t_i, q_i, v_i, u_i)\Delta\tau = 0, \quad i = 1, ..., N-1,\end{aligned} \tag{B22}$$

and

$$\frac{\partial L}{\partial u}(t_i, q_i, v_i, u_i)\Delta\tau + \gamma_i^{\mathrm{T}} \frac{\partial \mathrm{FD}}{\partial u}(q_i, v_i, u_i)\Delta\tau + \mu_i^{\mathrm{T}} \frac{\partial C}{\partial u}(t_i, q_i, v_i, u_i)\Delta\tau = 0, \quad i = 1, ..., N-1. \tag{B23}$$

Equations (B19)–(B22) are manipulated into

$$\lambda_N = \left\{ \left(\frac{\partial\delta}{\partial q^+} \right)^{\mathrm{T}} \right\}^{-1} (q_{N-1}, q_N) \left(\frac{\partial\varphi}{\partial q} \right)^{\mathrm{T}} (t_N, q_N, v_N), \tag{B24}$$

$$\lambda_i = \left\{ \left(\frac{\partial \delta}{\partial q^+} \right)^{\mathrm{T}} \right\}^{-1} (q_{i-1}, q_i) \left\{ - \left(\frac{\partial \delta}{\partial q^-} \right)^{\mathrm{T}} (q_i, q_{i+1}) \lambda_{i+1} + \left(\frac{\partial L}{\partial q} \right)^{\mathrm{T}} (t_i, q_i, v_i, u_i) \Delta\tau + \left(\frac{\partial \mathrm{FD}}{\partial q} \right)^{\mathrm{T}} (q_i, v_i, u_i) \gamma_{i+1} \Delta\tau + \left(\frac{\partial C}{\partial q} \right)^{\mathrm{T}} (t_i, q_i, v_i, u_i) \mu_i \Delta\tau \right\} = 0, \quad i = 1, ..., N-1, \tag{B25}$$

$$\gamma_N = \left(\frac{\partial \varphi}{\partial v} \right)^{\mathrm{T}} (t_N, q_N, v_N) + \left(\frac{\partial L_a}{\partial v^+} \right)^{\mathrm{T}} (v_{i-1}, v_N) \Delta\tau, \tag{B26}$$

and

$$\gamma_i = \gamma_{i+1} + \lambda_{i+1} \Delta\tau + \left(\frac{\partial L}{\partial v} \right)^{\mathrm{T}} (t_i, q_i, v_i, u_i) \Delta\tau + \left(\frac{\partial L_a}{\partial v^-} \right)^{\mathrm{T}} (v_i, v_{i+1}) \Delta\tau + \left(\frac{\partial L_a}{\partial v^+} \right)^{\mathrm{T}} (v_{i-1}, v_i) \Delta\tau + \left(\frac{\partial \mathrm{FD}}{\partial v} \right)^{\mathrm{T}} (q_i, v_i, u_i) \gamma_{i+1} \Delta\tau + \left(\frac{\partial C}{\partial v} \right)^{\mathrm{T}} (t_i, q_i, v_i, u_i) \mu_i \Delta\tau = 0, \quad i = 1, ..., N-1. \tag{B27}$$

ACKNOWLEDGMENTS

This work was partly supported by JSPS Kakenhi Grant Number JP15H02257.

References

1. Magni L, Raimondo DM, Allgöwer F. *Nonlinear Model Predictive Control: Towards New Challenging Applications*. 384. Springer . 2008.

2. Graichen K, Käpernick B. A Real-time gradient method for nonlinear model predictive control. In: Zheng T., ed. *Frontiers of Model Predictive Control* InTech. 2012 (pp. 9–28).

3. Ohtsuka T. A continuation/GMRES method for fast computation of nonlinear receding horizon control. *Automatica* 2004; 40(4): 563–574.

4. Diehl M, Bock HG, Schlöder JP. A real-time iteration scheme for nonlinear optimization in optimal feedback control. *SIAM Journal on Control and Optimization* 2005; 43(5): 1714–1736.

5. Todorov E, W Li . A generalized iterative LQG method for locally-optimal feedback control of constrained nonlinear stochastic systems. In: Proceedings of the 2005, American Control Conference, 2005. IEEE. ; 2005: 300–306.

6. Deng H, Ohtsuka T. A parallel Newton-type method for nonlinear model predictive control. *Automatica* 2019; 109: 108560.

7. Zanelli A, Domahidi A, Jerez J, Morari M. FORCES NLP: An efficient implementation of interior-point methods for multistage nonlinear nonconvex programs. *International Journal of Control* 2020; 93(1): 13-29.

8. Zhao J, Diehl M, Longman R, Bock H, Schlöder J. Nonlinear model predictive control of robots using real-time optimization. In: . 2 of *Collection of Technical Papers - AIAA/AAS Astrodynamics Specialist Conference*. AIAA/AAS. ; 2004.

9. Gerdts M, Henrion R, Hömberg D, Landry C. Path planning and collision avoidance for robots. *Numerical Algebra, Control & Optimization* 2012; 2(3): 437-463.

10. Tassa Y, Erez T, Todorov E. Synthesis and stabilization of complex behaviors through online trajectory optimization. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE/RSJ. ; 2012: 4906-4913.

11. Koenemann J, Del Prete A, Tassa Y, et al. Whole-body model-predictive control applied to the HRP-2 humanoid. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE/RSJ. ; 2015: 3346-3351.

12. Neunert M, Stäuble M, Giftthaler M, et al. Whole-body nonlinear model predictive control through contacts for quadrupeds. *IEEE Robotics and Automation Letters* 2018; 3(3): 1458-1465.

13. Carius J, Ranftl R, Koltun V, Hutter M. Trajectory optimization with implicit hard contacts. *IEEE Robotics and Automation Letters* 2018; 3(4): 3316-3323.

14. Budhiraja R, Carpentier J, Mastalli C, Mansard N. Differential dynamic programming for multi-phase rigid contact dynamics. In: 2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids). IEEE. ; 2018: 1-9.

15. Featherstone R. The calculation of robot dynamics using articulated-body inertias. *The International Journal of Robotics Research* 1983; 2(1): 13-30.

16. Featherstone R. *Rigid Body Dynamics Algorithms*. Springer . 2008.

17. Carpentier J, Mansard N. Analytical derivatives of rigid body dynamics algorithms. In: Robotics: Science and Systems. RSS. ; 2018: hal-01790971v2f.

18. Giftthaler M, Neunert M, Stäuble M, Frigerio M, Semini C, Buchli J. Automatic differentiation of rigid body dynamics for optimal control and estimation. *Advanced Robotics* 2017; 31(22): 1225–1237.

19. Erez T, Todorov E. Trajectory optimization for domains with contacts using inverse dynamics. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE/RSJ. ; 2012: 4914-4919.

20. Posa M, Cantu C, Tedrake R. A direct method for trajectory optimization of rigid bodies through contact. *The International Journal of Robotics Research* 2014; 33(1): 69-81.

21. Katayama S, Ohtsuka T. Efficient solution method based on inverse dynamics of optimal control problems for fixed-based rigid-body sysems. In: Proceedings of the 21st IFAC World Congress. IFAC. ; 2020.

22. Baumgarte J. Stabilization of constraints and integrals of motion in dynamical systems. *Computer Methods in Applied Mechanics and Engineering* 1972; 1(1): 1–16.

23. Murray. RM, Sastry SS, Zexiang L. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Inc. . 1994.

24. Bellicoso CD, Jenelten F, Fankhauser P, Gehring C, Hwangbo J, Hutter M. Dynamic locomotion and whole-body control for quadrupedal robots. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE/RSJ. ; 2017: 3359-3365.

25. Blanco JL. A tutorial on se(3) transformation parameterizations and on-manifold optimization. In: Technical report, University of Malaga. University of Malaga. Tech Repo; 2010.

26. Wieber PB. Holonomy and nonholonomy in the dynamics of articulated motion. In: Diehl M, Mombaur K. , eds. *Fast Motions in Biomechanics and Robotics: Optimization and Feedback Control*Springer. 2006 (pp. 411–425).

27. Stella L, Themelis A, Sopasakis P, Patrinos P. A simple and efficient algorithm for nonlinear model predictive control. In: 2017 IEEE 56th Annual Conference on Decision and Control (CDC). IEEE. ; 2017: 1939–1944.

28. Diehl M, Ferreau HJ, Haverbeke N. Efficient numerical methods for nonlinear MPC and moving horizon estimation. In: Magni L, Raimondo DM, Allgöwer F. , eds. *Nonlinear Model Predictive Control*. 384 of *Lecture Notes in Control and Information Sciences*. Springer. 2009 (pp. 391–417).

29. Richter SL, Decarlo RA. Continuation methods: Theory and applications. *IEEE Transactions on Systems, Man, and Cybernetics* 1983; SMC-13(4): 459–464.

30. Kelley CT. *Iterative Methods for Linear and Nonlinear Equations*. Society for Industrial and Applied Mathematics . 1995.

31. Carpentier J, Saurel G, Buondonno G, et al. The Pinocchio C++ library – A fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives. In: IEEE International Symposium on System Integrations (SII). IEEE. ; 2019: 614–619.

32. Carpentier J, Valenza F, Mansard N, others . Pinocchio: fast forward and inverse dynamics for poly-articulated systems. https://stack-of-tasks.github.io/pinocchio; 2015–2020.

33. Wächter A, Biegler L. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming* 2006; 106: 25-57.

34. Hutter M, Gehring C, Jud D, et al. ANYmal - a highly mobile and dynamic quadrupedal robot. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE/RSJ. ; 2016: 38–44.

35. Hwangbo J, Lee J, Hutter M. Per-contact iteration method for solving contact dynamics. *IEEE Robotics and Automation Letters* 2018; 3(2): 895–902.

36. raisimLib. https://github.com/leggedrobotics/raisimLib; .

37. raisimOgre. https://github.com/leggedrobotics/raisimOgre; .

38. Katayama S, Doi M, Ohtsuka T. A moving switching sequence approach for nonlinear model predictive control of switched systems with state-dependent switches and state jumps. *International Journal of Robust and Nonlinear Control* 2020; 30(2): 719-740.

39. Bryson AE, Ho YC. *Applied Optimal Control*. Taylor and Francis . 1975.

40. Carius J, Ranftl R, Koltun V, Hutter M. Trajectory optimization with implicit hard contacts. *IEEE Robotics and Automation Letters* 2018; 3(4): 3316–3323.